Engineering Contextual Information for Pervasive Multiagent Systems

Gabriella Castelli, Marco Mamei, Franco Zambonelli

Dipartimento di Scienze e Metodi dell'Ingegneria – Università di Modena e Reggio Emilia
Via Amendola 2 – 42100 Reggio Emilia – ITALY
{name.surname}@unimore.it

Abstract. Multiagent systems for mobile and pervasive computing should extensively exploit contextual information both to adapt to user needs and to enable autonomic behavior. This raises the problem of how to represent, organize, aggregate, and make available such data so as to have it become meaningful and usable knowledge, facilitating the design and development of agents, and enabling them to acquire high-degrees of context awareness at limited efforts. In this paper, we identify the key software engineering challenges introduced by the need of accessing and exploiting huge amount of heterogeneous contextual information. Following, we survey the relevant proposals in the area of context-aware pervasive computing, data mining and granular computing discussing their potentials and limitations. On these bases, we propose the W4 model for contextual data and show how it can represent an effective model to enable flexible general-purpose management of contextual knowledge, to facilitate agents in achieving high degrees of context-awareness and, overall, to facilitate the design and development of complex multiagent systems.

Keywords: Context-awareness, Autonomic services, W4 Model, Knowledge engineering.

1 Introduction

Pervasive and mobile computing scenarios consider the possibility of providing users with ubiquitous and on-the-move access to digital services, and of supporting users interactions with their surrounding environments [HenIR06, ManZ06, Cas07]. For this possibility to become practical and satisfying, agents should be able to understand situations occurring in the surrounding physical context, autonomously adapt their behavior to the context from which they are requested, and proceed with their execution in an autonomic (i.e., self-organizing, self-adapting, and self-healing) way [ManZ06]. The enforcement of these features requires both the technology to capture contextual data and the capability of agents to exploit it.

The technology to acquire contextual information is becoming increasingly available, and it will soon become widespread via the increasing deployment of RFID tags, sensor networks, localization systems, users' and organizations' profiles. This fact, together with the increasing success of participatory Web 2.0 tools, will

soon make available to agents overwhelming amounts of information about facts and events occurring in the physical and social world. This opens up the possibility of exploiting all such information for the provisioning of pervasive context-aware services for "browsing the world", i.e., for facilitating users in gathering information about the world, interacting with it, and understanding it. Those services require accessing to great amount of distributed and continuously updating data, however due to the amount of data and the inherent distribution the challenge is not getting the freshest data or all data available everywhere but getting a good approximation is real time. Accordingly, the real challenge for future pervasive applications is the investigation of principles, algorithms, and tools, via which this growing amount of distributed information can be represented, organized, aggregated, and made more meaningful, so as to facilitate the exploitation by agents [Bau06].

In the past few years, a number of different research communities, from pervasive computing, to data mining and granular computing, have recognized the above problems, and have proposed solutions aimed at engineering large amounts of contextual data and at turning them into usable knowledge [DeyA00, CegR06, DeGM05, Yao06a]. Thus, a first contribution of this paper is to survey such diverse proposals in different areas and critically analyze their potentials and limitations. The result of our analysis is that, despite the potentials of specific approaches for specific problems, none of them can qualify as a fully-fledged general-purpose solution for the challenges raised by pervasive autonomic multiagent systems. Following, the second contribution of this paper is to present a novel data model to represent context information, and discuss its potential to act as a general-purpose model to handle several kinds of context information. The model, which we call "W4", is based on the consideration that most information about the world (i.e., about facts occurring in the world) can be simply represented in terms of four "W"s – Who, What, Where, When. Despite its simplicity, such a representation enables for very expressive and flexible data usages. In particular, W4 data can be easily queried and accessed by agents, tolerate the effective execution of semantic data organization and data aggregation, and can be effectively used to represent both primitive data and high-level knowledge related to a situation.

2 Agents and Context-awareness

Our everyday environments (houses, offices, and cities) are increasingly populated by a variety of communication-enabled computing devices, forming the basis of a truly pervasive network and generating increasing amounts of information about the physical world and its processes. Embedded and wireless sensors collect and make available information about physical phenomena, RFID tags can be attached to objects to describe them and to track their usage. We, as humans, typically carry on a mobile phone and/or a PDA, possibly a GPS device and some additional wearable sensors, and can generate a lot of information about ourselves and about our own activities and movements. Similar considerations increasingly apply to cars and home furniture. In addition, the success of participatory Web tools (aka Web 2.0 technologies) and of geographic Web tools (e.g., Google Earth and alike), is

increasingly making available nearly-up-to-date information about various facts and events occurring in the physical and social worlds. The above trend paves the way for the design and development of a wide variety of innovative, autonomic and context-aware, pervasive multiagent systems. However, it also raises peculiar challenges in engineering such agents, mostly due to the issue of engineering all available data and turning it into usable knowledge.

2.1 Context-awareness vs. Situation-Awareness

Agents have to be inherently context-aware. In fact, collecting information about situations around and acting accordingly is the very core of their activities [ManZ06]. The need for context-awareness also arises when one wants to enforce autonomic behavior in agents, i.e., their capability of self-organizing their activities, self-reconfigure and self-heal on need. Given the intrinsic dynamics and decentralization of pervasive scenarios, where components and devices belonging to different stakeholders can come and go at any time and where the structure of the network is inherently dynamic and unreliable, autonomic behavior is necessary to ensure agents' continuity without forcing costly and hard to be managed human intervention. However, for such autonomic features to be enabled, agents require the capability of understanding what's happening around and react accordingly.

Given that, as stated at the beginning of this section, a number of technologies exists that contribute producing large amounts of contextual information, one may think that achieving context-awareness is simply a non issue. Whatever the data source producing some raw item of contextual information (i.e., "data atom"), all of them contribute populating a large cloud of data atoms and at making it available to agents (see Fig. 1-left). An agent in need of understanding what's happening around can access (i.e., internalize) the needed data atoms and analyze them.

Unfortunately, such description is far too simplistic and does not emphasize a number of complexities inherent in it. First, the process of data internalization can lead to high communication and computational costs for a multiagent system, in that it may require accessing large amounts of data atoms possibly distributed across different devices and analyze which data may serve its current purposes. Second, the process of analyzing retrieved contextual data atoms and turning them into useful knowledge may be non-trivial. In other words, getting access to context information does not automatically imply the capability of reaching "situation-awareness", i.e., the capability of recognizing a situation. That is, acquiring contextual information does not imply the capability of understanding situations, especially in the presence of an overwhelming amount of unrelated contextual data atoms. Such problem is even exacerbated by the increasing heterogeneity of devices and tools that contribute producing contextual information, and by the consequent need of handling heterogeneity in data representation and semantics.

In our view there must be an evolution from a model of simple context-awareness, in which agents access isolated pieces of contextual data and are directly in charge of digesting them, towards a model of situation-awareness, in which agents access properly structured and organized information, reflecting comprehensive knowledge that is related to a "situation" of interest and which can be exploited in a standardized fashion [Bau06]. With reference to Fig. 1-right, we

envision that the access by agents to contextual information does no longer occur directly, but rather via a “knowledge network” layer. Such layer should encapsulate mechanisms and tools to analyze and (self-)organize contextual information into sorts of structured collections of related data items, i.e. knowledge networks. Such knowledge networks, by pre-digesting contextual information and by providing compact and expressive information to agents, may support them in reaching, with reduced efforts, a comprehensive understanding of “situations” around and, consequently, a higher-degree of adaptability and autonomicity.

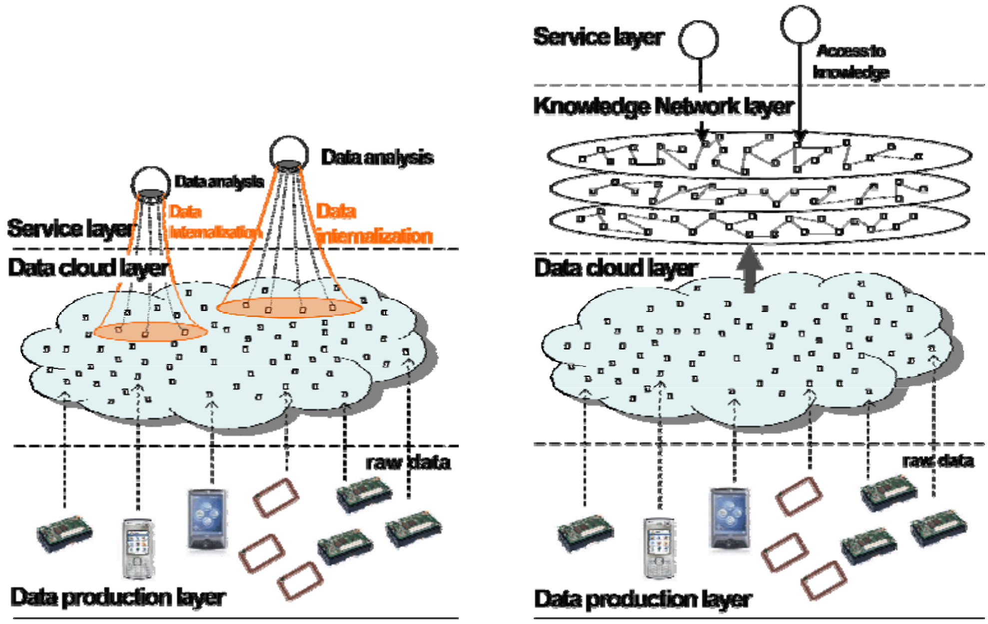

Fig 1: (left) Pervasive devices and sensors make available to agents a sort of “data cloud layer”, fed with large amounts of heterogeneous data atoms. To serve their purposes, a multiagent system needs to retrieve contextual information (i.e., internalize data atoms from the cloud), analyze it to properly understand situations, and finally exploit such knowledge as needed for their own goals. **(right)** By exploiting a knowledge network layer, agents are no longer forced to access the raw data cloud layer. Knowledge organization and analysis is externalized in the middleware, and agents are given access to pre-digested information.

From the software engineering viewpoint, an approach based on knowledge networks has the advantage of providing a clear separation of concerns between data analysis and data exploitation. While data analysis and organization is delegated to the knowledge network layer, agents are left with the only duty of exploiting such data to reach specific functionalities. As always in software engineering, this

separation of concerns can notably reduce the complexity of developing and maintaining agents. A possible criticism of the approach is that it does not eradicate the problem of analyzing large amounts of information, but simply passes it to a different component that either exists at application or at knowledge network level. However, in a distributed setting, knowledge networks can take care of knowledge management duties that would have been otherwise replicated inside each agent, with an overall saving in computation and communication efforts.

2.2 Engineering Challenges

For agents to effectively achieve situation-awareness, and for our general idea of knowledge networks to become a practical tool, several engineering challenges have to be faced.

Data Model. Our idea of knowledge networks requires the identification of a simple, general-purpose, and uniform model to represent contextual information. The model should enable representing very diverse facts about the context, typically generated by a variety of heterogeneous sources and at different levels of granularity. Also, the model should enable ease of manipulation of the data atoms, both by the algorithms and that will be embedded in knowledge networks and by agents. Furthermore, the model should enable to deal with incomplete information and information of limited accuracy.

Access to Data. The very goal of knowledge networks is to provide knowledge to agents and to digest data from any possible contextual data source. First, it is necessary to identify suitable methods (i.e., APIs) by which agents can be given access to the knowledge network layer and the information within. Such methods should enable flexible querying of the knowledge network layer, yet should limit the amount of information required by agents to fruitfully access it. Also, given the intrinsic distributed and decentralized nature of contextual information, access to knowledge networks by multiagent systems should abstract from the actual distribution and allocation of data and knowledge. Second, a similar general API should be provided for enabling data sources (which may include agents themselves) to inject new data in the knowledge network layer.

General Approaches for Data Aggregation and Networking. Our idea of knowledge networks is to have it as a “live” layer, which is in charge of continuously and autonomously analyzing information to aggregate data atoms, relate existing knowledge atoms with each other, and extract meaningful knowledge from the available data. Thus, it will be necessary to identify algorithmic approaches for performing such analysis without explicit human intervention. Also, it is necessary for such algorithmic approaches to be as general as possible, so as to be flexibly adaptable to a variety of application needs without having to re-think from scratch the knowledge network architecture for any new application needs.

Application-specific Views. Strictly related to the above, we have clear in mind that the idea of a single knowledge network capable of capturing of the possible needs of all possible agents, is illusionary. In a given pervasive environment, a variety of agents by different stakeholders may exist and new ones can be deployed at any moment, each with its own goals. Such agents may, of course, have very diverse needs for what concerns to accessing contextual knowledge, and may

require organization/aggregation along different dimensions and based on different algorithms. For instance, some agents may be more interested in organizing/analyzing knowledge along the spatial dimension (e.g., for detecting spatial patterns of activities in an environment), some along the temporal dimension (e.g., for detecting temporal patterns of activities by users), some along some mixed spatio-temporal dimension or along any other application-specific dimension. Thus, agents may require the dynamic instantiation within the knowledge networks of application-specific algorithms for knowledge analysis. Accordingly, a general approach to knowledge networks should account for this possibility and enable the dynamic instantiation of any new knowledge analysis algorithm.

3 Critical Survey of Relevant Related Works

In the last years the role of the environment has being rapidly considered as a fundamental one in modeling and engineering multi-agent systems. The environment abstraction suits to some extent our idea of knowledge network. For this perspective, the knowledge network can be perceived as an information environment where agents live.

[Ban02] provides an explicit representation of the spatial structure of the environment. They define a Multilayered Multi–Agent Situated System (MMASS) that describes the spatial structure of the environment as a multilayered network of sites. Agents diffuse fields throughout the environment, and since fields values can decrease during propagation, agents perceive them depending from where their position in space. The work in [Vir05] interprets the environment as a locus to be explicitly designed and developed to provide agents with services. They introduce the notion of artifact for MAS, i.e. entities residing in the environment independently of the existence of agents. The artifact exposes a set of operations, which an agent aware of the artifact can invoke. [Chan05] proposes a cognitive middle layer, starting from the idea that agents must be able to understand the environment and capture its dynamic nature. It is realized as a three-layered architecture. The bottom layer is the physical environment. The middle layer is the concept model, it is merged into the environment and shared among all agents providing a common conceptual basis. And finally the topmost layer is the subjective mind, which resides in an agent. [Sche06] considers the problem of distributed mobile application where the interactions between agents is complicated by the dynamics of the environment. They propose a distributed interaction protocol based upon roles called ObjectPlaces.

All these approaches provides interesting models that could be applied to our knowledge network idea. However, even if the knowledge network base infrastructure could be based on environment-based modes, the fact that we are dealing explicitly with context acquisition and processing requires dedicated models and algorithms. In the rest of this section, we survey and analyze several approaches that, in different areas, are somewhat contributing pieces towards the realization of our knowledge network vision. Yet, none of the approaches we have analyzed properly addresses all the identified challenges.

3.1 Context-awareness

In the past few years, a lot of research work has focused on identifying proper models for context-aware information, mainly with the goals of engineering usable context representation from low-level sensor data.

Early work in this area, as from Schmidt et al. [Sch99] and Dey et al. [DeyAS01], concentrates on the issue of acquiring context data from sensors and of the processing such data to make it available to processes/agents in the form of abstract components. Such approaches thus partially address the data model challenge, in that they recognize the need for a data model able to abstract raw contextual data at higher level. However, they generally miss in identifying a uniform model and a common semantics to describe the data. This forces developers to build new query languages and new components in dependence of the kind of information at hand.

A different thread of researchers focuses on the issue of providing rich data models for contextual information and of facilitating querying by agents. [SchAW94] proposes a context model in which each contextual data atom is represented by a set of environments variables – each one dealing with a different aspect of the context – that can be accessed in a flexible way. Similarly, [HenIR06] models contextual data by making possible to enrich it with several meta-information such as temporal aspects, information imperfection, etc. Such approaches well address the data model challenge, in that they aim at developing a general and flexible data model. However, working with data atoms made up of a long list of elements/variables does not go toward simplicity and generality, which we instead feel should both be goals. More recently, other proposals have adopted a similar endeavor but have considered the issue of adopting specific ontologies to model context-information and enable – other than efficient querying – also efficient context-reasoning [Chen04, Rou06, LeeM07]. Although such approaches tend to be application-specific, they attribute the importance of linking independent atoms of contextual information (with ontological relations) and of reasoning not only on individual data items but also on their relations, an idea which is fully shared by our knowledge network vision.

An increasing number of research work get inspiration from tuple space data models [AhuCG86] and proposes representing contextual information in the form of tuples, storing them across a set of distributed tuple spaces holding local contextual information, and accessing them via associative (i.e., pattern-matching based) query operations on tuple spaces. Egospaces [JulR06] adopts this perspective, without committing to a specific pre-defined structure for context tuple, which can make it difficult for agents to uniformly deal with tuples representing different aspects of the context represented in different formats. However, Egospaces proposes a so called "egocentric" notion of context, in which different agents can perceive a different context-dependent representation of the contextual information, depending on their current location. We consider such a feature very important in that it allows to tailor information to specific users' needs and viewpoints. The Context Fabric model [Hon02] improved the Egospaces idea in that it relies on well-structured context tuples each describing a single piece of context data in terms of entities (people, place, thing), attributes (e.g., the name). Moreover, even if it does not propose solutions for enforcing application-specific views, it considers the possibility of

identifying relationships between context tuples. Recent proposals focusing on sensor networks, suggest exploiting a tuple-based approach to flexibly access sensorial information [NewW04]. The idea is to have agents inject code into the sensors for aggregating/elaborating data within the network, and eventually enabling agents to directly access aggregated data according to their own specific needs.

In general we consider tuple-based approaches very suitable for organizing and accessing contextual information, but we also think that there is need of more structuring and flexibility than those exhibited by the existing approaches.

Some recent proposals focus on providing models for contextual data that adopt a uniform well-defined structure, capturing those specific aspects which are of interest by agents, a characteristic which we consider very promising. The approach proposed in the Nexus platform [Leh04] proposes representing different contextual data uniformly accounting for fields such as spatial references and temporal references, for enabling general spatial and temporal queries over a context database. Similarly, the proposal described in [XuC05] suggests adopting a seven-field data structure to describe the context. The suggested fields include subject, predicate, object, time, area, certainty, freshness, which overall provide quite a complete characterization of contextual information. Finally, the system described in [Bra06] proposes describing contextual information contained in RFID tags in terms of identity, purpose of tag, location of tag, time of information production. Although the system is a special-purpose one, having been applied to RFID tags only, we consider it interesting in that is consider a simple enough yet quite informative structured data model, able to represent in a uniform way different data coming from different sources. Indeed, our proposal accounts for a very similar structuring for contextual information, and enriches it with a well defined API, and with the possibility of linking data atoms and of providing application-specific views.

3.2 Data Mining & Pattern Discovery

As stated in Sect. 2.1, the potentially overwhelming amount of data that can be generated by pervasive sensing infrastructures does not constitute knowledge per se, in that agents have may have to face complex analysis tasks to get a meaning out of it. Such analysis task, which we think should be delegated to a knowledge network layer, is in the end a sort of data mining process [GalS06].

Data mining concerns analyzing large amounts of typically unrelated data to infer hidden linkages, correlations, rules, constraints, (i.e., broadly speaking, patterns) in such data [Fay96, Bar03], to present such inferences to user for subsequent interpretation, i.e., to have the user give a meaning to data by analyzing the identified patterns. In general, all the mechanisms proposed in this field can be naturally be employed within the knowledge network layer to extract knowledge from raw data collected by sensing devices.

In the wide data mining research area, a variety of algorithms and approaches have been proposed, most of which rely on a general two step process: identifying relevant sets of related data item within the global dataset and, following, inferring patterns from this sets. This two-steps process can be of inspiration for knowledge networks, and in particular with regard to the need of identifying flexible and general algorithmic approaches for continuously and autonomously aggregate and

analyze data atoms. In fact, it may lead to a modular algorithmic approach, in which the two issues of relating data atoms and of extracting higher-level knowledge form such relations can be clearly separated.

Data mining activities may, in general, identify thousands of patterns from data sets, all of them of general interest. With this regard, several researches involve the specialization of association mining fundamentals to address the problems of specific application domains, e.g. spatial or temporal association rules, to limit the number of mined association to the most relevant ones [AlCG07, VerC06]. Of course, for applying the lessons of data mining to agents and knowledge networks, a similar application-specific approach must be taken.

Recently many researchers applied data mining techniques to wireless sensor networks. Sensor network offers new challenges to classical data mining. The large amount of sensed data has to be modeled has a stream, there may be a large number of nodes in the network, calling for decentralized approaches and should account for data losses, and finally the power consumption issue must be considered. Some approaches[BonB05, McS05] focuses on mining sensed data for prediction purpose. [BonB05] proposes a framework for data mining upon sensor network for supervised learning (prediction, classification, etc.) according to a specified level of precision and quality. The framework is based on a two step process: the first step performs aggregation of sensors in clusters, in the second one, each cluster sends the aggregate to a data mining server that performs the analysis. Similarly, [McS05] proposes a framework for prediction based on the flow of local predictors through the network. At the root, predictors are combined via a voting mechanism. Other approaches [GanEH04, KulD05] focuses on the general problem of identification of pattern by using neural networks algorithm in a distributed setting. In particular, these approaches: (i) uphold the need for data mining for analyzing the vast amount of data in pervasive computing application, (ii) show that decentralized approaches are effective and operable in distributed network with several nodes.

As an additional note, typical data mining approaches are human-centered and query-based, i.e., assuming humans are the end users of data mining activities. In pervasive computing, instead, we require automatic methodologies for discovering relations between contextual data and for making these available to computational agents. Moreover, conventional data mining assumes the independence between the attributes and the independence between the values of these attributes, but in context-awareness diverse attributes and their corresponding values are often related. Some researches has been done in areas where correlation between attributes exist [GalS06], however they result in human-centered and visual data mining methodologies that are not suitable for pervasive multiagent systems.

3.3 Data Aggregation Granular Computing

Granular computing is an emerging inter-disciplinary research area that considers the general issue of processing "information granules", i.e., collections of data atoms, and extract knowledge from them. The idea is to organize information granules together based on their similarity, functional properties, spatial/temporal adjacency, or identified regularities in data, and eventually provide higher knowledge-level views, at different scales, of the phenomena underlying

information granules. Although the strict relations between data mining and granular computing are evident, the latter adopts a more theoretical and inter-disciplinary viewpoint, and specifically focuses on the idea that, at different level of observation and analysis, the same data can provide different knowledge [Yao06a].

The ideas and principles of granular computing have been investigated in many research fields from computer science to psychology: computational intelligence, artificial intelligence, the theory of hierarchy, divide and conqueror, the theory of small groups, etc. Such a wide range of researches demonstrates the potential of granular computing approach. Indeed, the underlying assumption of granular computing is that the basic principles and methodologies are independent from specific problem domains. Granular computing naturally fits also our perspective on context-awareness and situation-awareness, and specifically our idea of knowledge networks. First, the goal of extracting knowledge from information granules directly maps into the idea of introducing a layer above the cloud of contextual data to access higher-level information. Second, the idea of relating information granules based on different characteristics and rules directly maps into our idea of knowledge networks. Finally, the idea of granular computing of providing multi-level views for serving different purposes directly maps into our goal of providing application-specific views of knowledge networks.

To the best of our knowledge, there are no studies directly related to applying granular computing ideas to support context-awareness by pervasive multiagent systems. Nevertheless, there are studies related to applying granular computing techniques to model spatial and temporal data at different levels of granularity, an issue which is of specific relevance to pervasive multiagent systems (which are inherently situated in space and time). Camossi et. al. [Cam06] proposes a spatiotemporal data model relying on an extended ODMG model, which provides a uniform management of both moving entities and temporal maps. The model allows for the multi-level management of such data, and also deals with temporal indeterminacy and spatial inaccuracy. Granular GeoGraph [DeGM05] provides a conceptual spatial data model with two granularity dimensions: a purely spatial one and a semantic one. Spatial granularity refers the possible variations of the geometry of an object with respect to different scales, semantic granularities refers to the possible variations of set of domain objects with respect to the levels of detail requested by different users/applications.

Although the above researches help us providing some inspiration with spatial and temporal data, context-awareness and situation-awareness especially, involves more contextual factors and more rich set of relations to be taken into account, something which we indeed try to account in our proposal.

4 The W4 Approach

The result of the previous survey is that, despite diverse approaches address specific engineering challenges, none of them propose fully-fledged solutions for the need of modern autonomic pervasive multiagent systems.

Our proposal for a novel, simple yet effective, data model for expressing contextual knowledge about the world starts from the consideration that any elementary data atoms as well as any higher-level piece of contextual knowledge, in the end, represents a “fact” which has occurred. Accordingly, our proposal simply account that any of such facts – and therefore any data/knowledge atom – can be expressed by means of a simple yet expressive 4-fields tuples (Who, What, Where, When): “someone or something (Who) does/did some activity (What) in a certain place (Where) at a specific time (When)”.

W4 knowledge atoms may be created by proper software agents associated to data sources or sensor. Their four-fields structure is flexible and general enough to uniformly deal with information coming from sources as diverse as embedded devices, cameras, users, or Web 2.0 sites, and can account for adaptation to context and incomplete information (i.e., some of the four fields being unspecified). W4 knowledge atoms, as tuples in tuple spaces, can be stored in suitable shared data spaces, whatever distributed and implemented. Users and agents, from everywhere, can retrieve knowledge atoms via a simple API, based on “à la Linda” [AhuCG86] pattern-matching query mechanisms. Such API supports context-aware queries and incomplete information, to enable agents to interact with the world and to enforce autonomic and context-aware functionalities. In addition, the simple W4 structure support general distributed algorithms for data aggregation and manipulation, and facilitates the building of semantic knowledge networks and of multiple, application-specific views.

4.1 Data Representation

The four-fields (Who, What, Where, When) of the W4 data model each describes a different aspect of a contextual fact.

The Who field associates a subject to a fact, and may represent a human person (e.g., a username) or an unanimated part of the context acting as a data source (e.g., the ID of an RFID tag). The Who field is represented by a type-value pair, in the form of a string, with an associated namespace that defines the “type” of the entity that is represented. For example, valid entries for this field are: “person:Gabriella”, “tag:tag#567”.

The What field describes the activity performed by the subject. This information can either come directly from the data source (e.g., a sensor is reading a temperature value), or be inferred from other context parameters (e.g., an accelerometer on a PDA can reveal that the user is running), or it can be explicitly supplied by the user. This field is represented as a string containing a predicate:complement statement. For example, valid entries for the What field are: “read:book”, “work:pervasive computing group”, “read:temperature=23”.

The Where field associates a location to the fact. In our model the location may be a physical point represented by its coordinates (longitude, latitude), a geographic region (we currently adopt the PostGIS language to describe such regions), or it can also be a logical place. In addition, context-dependent spatial expressions like “here” or “within:300m” can be used for context-aware querying, as described in the following of this section.

The When field associates a time or a time range to a fact. This may be an exact time/time range (e.g., “2006/07/19:09.00am - 2006/07/19:10.00am”), or a concise description (e.g., 9:28am). For example 9:28am = 2006/07/19:9:28am ± 5min. Also in this case, context-dependent expressions can be defined (e.g., “now”, “today”, “yesterday”, “before”) and can be used for context-dependent querying.

In summary, in our current implementation, the content of each W-field is a string of formatted text containing either some keywords like “yesterday”, “within”, etc., or some general unformatted words “read”, “work”, “temperature”, etc.

While the 4 Ws structure the information contained in a knowledge atom meaningfully, the content of each field is still difficult to be analyzed and, in general, to an agent something like “What = read:book” has the same meaning of “What = djhxf:wyktx”. In our experiments and applications, this problem is trivially solved by using a predefined small ontology hardcoded into the agent and enabling the agent to recognize specific words. This of course, while vey simple to implement, presents all sorts of problems with regard to openness and scalability of the knowledge network. In any case, this kind of problem (i.e., the need to use shared ontologies) is not peculiar of our approach, and it troubles all open systems. Accordingly, in our future work, we plan to describe the content of each W-field by making use of well-defined ontologies supporting interoperability between agents also in open and large scenarios.

4.2 Data Access and Multiagent system Engineering

As already stated, it is fundamental to define a simple API for agents to access to contextual knowledge and enabling data sources and agents to inject new data in the knowledge network layer. Since knowledge atoms are stored in the form of W4 tuples in a shared data space (or in multiple data spaces), we took inspiration from tuple-space approaches to define the following API:

```
void inject(KnowledgeAtom a);
KnowledgeAtom[] read(KnowledgeAtom a);
```

The inject operation is equivalent to a tuple space “in” operation: an agent accesses the shared data space to store a W4 tuple there.

The read operation is used to retrieve tuples from the data space via querying. A query is represented in its turn as a W4 tuple with some unspecified or only partly specified values (i.e., a template tuple). Upon invocation, the read operation triggers a pattern matching procedure between the template and the W4 tuples that already populate the data space. A vector of all matching tuples – i.e., those for which all the defined fields match those provided in the template – is returned as the result of the query. In any case, pattern matching operations work differently from the traditional tuple space model. In fact, our proposal relies on the W4 structure to enforce more expressive pattern matching operations, which may exploit differentiated mechanisms for the various W4 fields. Current mechanisms work as follows:

Who and What. Pattern-matching operations in these fields are based on string-based regular expressions. For example, “user:*” will match any user.

Where. Pattern matching in this field involves spatial operations inspired by PostGIS operations. Basically, the template defines a bounding box (e.g., “circle, center(lonY,latX), radius:500m”) and everything within the bounding box matches the template. All tuples with a Where field within the circle will match this field of the template. Contextual places such as “within:300m” can be specified in the template and are translated into actual spatial regions – based on the current location from where the query is performed – before going through the pattern matching.

When. In this case, the template defines a time interval. Everything that happened within that interval matches the template. Concise time descriptions as well as contextual ones (e.g., “now” or “before”) are converted into actual time intervals before pattern matching.

Two simple examples follow to illustrate the querying process. Let us assume Gabriella is walking in the campus and wants to know if some colleagues are near. She will ask (via a read operation):

```
Who: user:*
What: works:pervasive computing group
Where: circle,center(lonY,latX),radius:500m
When: now
```

Then, she will get in return the tuples representing all the colleagues of her group currently around (at least, of all those colleagues having decided to expose themselves via a W4 tuple). Similarly, Gabriella can ask if some of her colleagues have gone to work in the morning:

```
Who: user:*
What: works:pervasive computing group
Where: office
When: 2006/07/19:09am- 2006/07/19:10am
```

We emphasize that the returned answers have not to be “complete” W4 tuples. The pattern matching mechanism also allows for matches between incomplete information. Thus, unlike in traditional tuple space approaches, applications are based on components entering complete and incomplete context information and getting in response refined (but possibly still incomplete) information.

In summary, the proposed data access model reflects standard tuple-space operations, but can rely on a predefined structure in the tuples to support more meaningful and semantic kind of pattern matching.

4.3 Data Generation

In the W4 model, we rely on the reasonable assumption that software drivers (or, more in general, software agents) are associated with data sources and are in charge of creating W4 tuples and inserting them in some sorts of shared data spaces. In the end, any data source must be somehow associated with some software to gather and store data items, W4 agents have the additional goal of collecting all the necessary

information to produce a W4 tuple which is as accurate and complete as possible. This occurs by sensing and inferring information from all the devices and sources available (e.g., RFID tags, GPS devices, Web agents), and by combining them in a W4 tuple. Three simple examples may clarify this concept.

Let us assume Gabriella is walking in the campus park. Agents running on her GPS-equipped PDA, can periodically create the following tuple:

Who: user:Gabriella What: walk:4km/h Where: lonY, latX When: 2006/10/17:10.59am

Where the Who is entered implicitly by the user at the login, What and Where can be derived by the GPS (e.g., the speed of Gabriella as measured by the GPS can be used to deduce that she is walking), When can be provided both by the PDA or by the GPS. Viewing this from a different, more fine-grained perspective, we can imagine that one agent controlling the user profile can create a raw W4 tuple in which only the who and where are specified; another agent controlling the GPS agent create a tuple in which only where and what (i.e., the speed) are specified. Accordingly, the merging of these two raw W4 atoms into the complete one represented below can be considered as an action of the knowledge networking that produces a more complete and expressive information.

Now, let us assume that Gabriella's PDA is connected with a RFID tag reader. A specific RFID agent controls the reader and handles the event of "tag recognition" whenever a tag enters in the reading range. In this case, either the tag contains its own Who and What description in its limited memory, or the tag ID can be resolved in a database (mapping tag IDs into the associated Who-What descriptions) that the agent may access to fill in the W4 fields. Otherwise, the Who reduces to the tag ID (which enables to access to the database later) and the What is left empty. As in the previous example, the Where and When can be read from the GPS of the user. The resulting tuple is as follows:

Who: tag:#456 What: - Where: lonY, latX When: 2006/10/17:10.59am

The agent running in the knowledge network can use both the data coming from the GPS and the tag to provide a better localization of Gabriella. For example a good policy is that the RFID based location may be more accurate than the GPS one. So the resulting tuple describing Gabriella is the result of the merging between the previous ones:

Who: user:Gabriella What: walk:4km/h Where: tag#456 When: 2006/10/17:10.59am

This last example shows again a task of knowledge networking, in that it includes and action for relating individual atoms to increase their informative values.

5 Conclusion and open issues

The W4 Model is our proposal for expressing contextual knowledge about the world. It tackles the majority of challenges in Sect. 2.2. However we are still working on it to extend the aggregation mechanism and to test them in distributed environments. Its simple four-fielded structure can uniformly represents data coming from diverse sources, it can represent simple data atoms as well as aggregated atoms. The examples in Sect. 4 shows the expressiveness of the data model in diverse situations. The developed API to access the knowledge network layer and to inject new data is simple yet flexible in that it is based upon the classical tuple spaces mechanism, the query interface is based on expressive pattern matching upon the four fields. Respect to first works in the field of Context-awareness such as [Sch99] and [DeyAS01], the W4 Model can uniformly deal with multiple context information in a coherent way, without leading to a long list of all the characteristics of the context as [SchAW94] and [HenIR06]. Differently from the tuple based approaches ([JulR06] , [Hon02]) the W4 representation strongly structures the context representation, so that the context representation can be easily browsed. The W4 Model represents the context similarly to [Leh04], [XuC05] and [Bra06], but our approach is general purpose and able to represent a large number of context information.

About the knowledge network, our idea is to exploit the four fielded structure to identify some preferential dimensions between atoms, and create new atoms represented the inferred knowledge. In this, the proposed approach is similar to the classical data mining process, in which in a first step is devoted to identify all data sets, and then patterns and rules are inferred from sets.

Although powerful some problems and limitations affects the current W4 Model. The first criticism of the W4 approach is that it does not eradicate the problem of analyzing large amount of data, but simply passes it to a different abstraction level. It should be considered that knowledge network can take care of knowledge management duties that would have been otherwise replicated inside each agent.

A serious limitation of our model is the lacks of meta data about the context, such as the freshness of the data, the source of the data, etc. that are traditionally available to the agents. We plan to tackle this requirements in the future works.

Another problem deals with the storage of historical information. Although historical data are useful to do inferences, for learning procedures and in general for querying, it is not possible to store all W4 atoms, some mechanism to aggregate or delete old data must be developed. Moreover it must considered that the knowledge network leads to a multiplication of new atoms coming from the inferences process.

In our future work we will try to tackle all these challenges to finally develop flexible and autonomic knowledge networks.

Acknowledgements

Work supported by the project CASCADAS (IST-027807) funded by the FET Program of the European Commission.

References

[AhuCG86] Ahuja S., Carriero N., Gelernter D.: Linda and Friends, IEEE Computer, 19(8), 26--34, (1986).

[AlCG07] Al-Naymat G., Chawla S., Gudmundsson J.: Dimensionality reduction for long duration and complex spatio-temporal queries. In: Proceedings of the 2007 ACM symposium on Applied computing, pp. 393—397. ACM Press, Seoul(2007).

[Ban02] Bandini, S., Manzoni, S., and Simone, C.: Dealing with space in multi--agent systems: a model for situated MAS. In: Proceedings of the First international Joint Conference on Autonomous Agents and Multiagent Systems: Part 3, pp. 1183—1190, ACM Press, New York(2002).

[Bar03] Bartolini I., Bertino E., Catania B., Ciaccia P., Golfarelli M., Patella M., Rizzi S.: PAtterns for Next-generation DAtabase systems: preliminary results of the PANDA project. In: Symposium on Advanced Database Systems, pp. 293—300. Rubettino Ed., Cetraro (2003).

[Bau06] Baumgarten M., Bicocchi N., Mulvenna M., Zambonelli F.: Self-organizing Knowledge Networks for Smart World Infrastructures. In: International Conference on Self-organization in Agents. Erfurt(2006).

[BonB05] Bontempi G., Le Borgne Y.: An adaptive modular approach to the mining of sensor network data. In: Proceedings of 1st International Workshop on Data Mining in Sensor Networks as part of the SIAM International Conference on Data Mining, pp. 3—9. SIAM Press, Newport Beach (2005).

[Bra06] Bravo J., Hervas R., Chavira G., Nava S.: Modeling Contexts by RFID-Sensor Fusion. In: Conference on Pervasive Computing and Communications Workshops, pp. 30--34 . IEEE Computer Society Press, Pisa (2006).

[Cam06] Camossi E., Bertolotto M., Bertino E.: A flexible Approach to Spatio-temporal Multigranularity in an Object Data Model. In International Journal of Geographical Information Science, 20(5), 511-534 (2006).

[Chan05] Chang P. H.-M. , Chen K.-T. , Chien Y.-H., Kao E., Soo V.-W.: From Reality to Mind: A Cognitive Middle Layer of Environment Concepts for Believable Agents. In: Environments for Multi-Agent Systems LCNS, vol. 3374, pp. 57—73. Springer, Heidelberg (2005).

[Cas07] Castelli G., Rosi A., Mamei M., Zambonelli F.: A Simple Model and Infrastructure for Context-Aware Browsing of the World. In: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications, IEEE Computer Society Press, New York (2007).

[CegR06] Ceglar A., Roddick J. F.: Association Mining. In: ACM Computing Surveys, 38(2) (2006).

[Chen04] Chen H., Perich F., Finin T., Joshi A.: SOUPA: StandardOntology for Ubiquitous and Pervasive Applications. In: Proceedings of 1st Annual International Conference on Mobile and Ubiquitous Systems

(MobiQuitous2004), Networking and Services, IEEE Computer Society Press, Boston (2004).

[DeGM05] De Fent I., Gubiani D., Montanari A.: Granular GeoGraph: a Multi-Granular Conceptual Model for Spatial Data. In: Proceedings of the Thirteenth Italian Symposium on Advanced Database, Aracne Ed., Bressanone, (2005).

[DeyAS01] Dey A. K., Abowd G. D., Salber D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications. In: Human-Computer Interaction, 16(2-4), :97—166 (2001).

[Fay96] Fayyad U.M., Piatetsky-Shapiro G., Smith P.: Advantages in knowledge discovery and data mining. In: Data Mining to knowledge Discovery: An Overview, AAAI/MIT Press, 1996.

[GalS06] Galloway J., Simoff S. J.: Network Data Mining: Methods and Techniques for Discovering Deep Linkage between Attributes. In: Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling, Australian Computer Society Inc., Hobart (2006).

[GanEH04] Ganesan D., Estrin D., Heidemann J.: DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks?. In: ACM SIGCOMM Computer Communication Review, 33(1), 143—148 (2003).

[HenIR06] Henricksen K., Indulska J., Rakotonirainy A.: Developing Context-aware Pervasive Computing Applications: Models and Approach. In: Journal of Pervasive and Mobile Computing, 2(1), 37--64 (2006).

[Hon02] Hong J.: The Context Fabric: An Infrastructure for Context-Aware Computing. In: Conference on Computer Human Interaction, Minneapolis (2002).

[JulR06] Julien C., Roman G.: EgoSpaces: Facilitating Rapid Development of Context-aware Mobile Applications. In: IEEE Transactions on Software Engineering, 32(5), 281--298 (2006).

[KulD05] Kulakov A., Davcev D.: Data mining in wireless sensor networks based on artificial neural-networks algorithms. In: Proceedings of 1st International Workshop on Data Mining in Sensor Networks as part of the SIAM International Conference on Data Mining, pp. 10 – 16, SIAM Press, Newport Beach (2005).

[LeeM07] Lee D., Meier R.: Primary Context Model and Ontology: A Combined Approach for Pervasive Transportation Agents. In: Proceedings of the 5th IEEE Conference on Pervasive Computing and Communications Workshops, IEEE Computer Society Press, White Plains (2007).

[Leh04] Lehmann O., Bauer M., Becker C., Nicklas D.: From home to world - supporting context-aware applications through world models. In: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications, pp. 297, IEEE Computer Society Press, Orlando (2004).

[ManZ06] Manzalini A., Zambonelli F.: Towards Autonomic and Situation-Aware Communication Agents: the CASCADAS Vision. In:

Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, pp. 383—388, IEEE Computer Society Press, Prague (2006).

[McS05] McConnell S. M., Skillicorn D. B.: A Distributed Approach for Prediction in Sensor Networks. In: Proceedings of 1st International Workshop on Data Mining in Sensor Networks as part of the SIAM International Conference on Data Mining, pp. 28—37, SIAM Press, Newport Beach (2005).

[NewW04] Newton R., Welsh M.: Region Streams: Functional Macroprogramming for Sensor Networks. In: Proceedings of the 1st international workshop on Data management for sensor networks, pp. 78—87, ACM Press, Toronto (2004).

[Rou06] Roussaki I., Strimpakou M., Pils C., Kalatzis N., Anagnostou M.: Hybrid context modeling: A location-based schemeusing Ontologies. In: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops, IEEE Computer Society Press, Pisa (2006).

[Sche06] Schelfthout, K., Weyns, D., and Holvoet, T.: Middleware for Protocol-Based Coordination in Mobile Applications. IEEE Distributed Systems Online 7, (2006).

[SchAW94] Schilit B., Adams N., Want R.: Context-Aware Computing Applications. In: Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 85—90, IEEE Computer Society Press, Santa Cruz (1994).

[Sch99] Schmidt A., Aidoo K. A., Takaluoma A., Tuomela U., Van Laerhoven K., Van de Velde W.: Advanced Interaction in Context. In: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pp. 187—201, Springer-Verlag, Karlsruhe (1999).

[VerC06] Verhein F., Chawla S.: Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. In: Proceeding of the 11th International Conference on Database Systems for Advanced Applications, Springer Link, Singapore (2006).

[Vir05] Viroli M., Omicini A., Ricci A.: Engineering MAS environment with artifacts. In: 2nd International Workshop Environments for Multi-Agent Systems, Utrecht(2005).

[XuC05] Xu C., Cheung S. C.: Inconsistency Detection and Resolution for Context-aware Middleware Support. In: Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering, ACM Press, Lisbon (2005).

[Yao06a] Yao Y.Y.: Three perspectives of granular computing. Journal of Nanchang Institute of Technology, 25(2), 16-21 (2006).